ACEEE Proc. of Int. Conf. onComputational Intelligence and Information Technology

Indexing based Genetic Programming Approach to Record Deduplication

Natasha R. Shaikh
Marathwada Institute of Technology, Aurangabad, India
Email: shaik.natasha@gmail.com

***Abstract*—In this paper, we present a genetic programming (GP) approach to record deduplication with indexing techniques.Data de-duplication is a process in which data are cleaned from duplicate records due to misspelling, field swap or any other mistake or data inconsistency. This process requires that we identify objects that are included in more than one list.The problem of detecting and eliminating duplicated data is one of the major problems in the broad area of data cleaning and data quality in data warehouse. So, we need to create such a algorithm that can detect and eliminate maximum duplications.GP with indexing is one of the optimization technique that helps to find maximum duplicates in the database. We used adeduplication function that is able to identify whether two or more entries in a repository are replicas or not. As many industries and systems depend on the accuracy and reliability of databases to carry out operations. Therefore, the quality of the information stored in the databases, can have significant cost implications to a system that relies on information to function and conduct business. Moreover, this is fact that clean and replica-free repositories not only allow the retrieval of higher quality information but also lead to more concise data and to potential savings in computational time and resources to process this data.**

***Index Terms*—genetic programming, indexing, de-duplication, optimization, standardization, data cleaning**

I. INTRODUCTION

There is an increasing demand for systems that canprovide secure data storage in a cost-effective manner.Havingduplicate records occupies more space and evenincreases the access time. Thus there is a need toeliminate duplicate records**.**The increasing volume of information available in digital media has become a challenging problem for data administrators. Usually built on data gathered from different sources, data repositories such as those used by digital libraries and e-commerce brokers may present records with disparate structure. Also, problems regarding low-response time, availability, security, and quality assurance become more difficult to handle as the amount of data gets larger.Today, it is possible to say that the capacity of an organization to provide useful services to its users is proportional to thequality of the data handled by its systems.In this environment, the decision of keeping repositories with "dirty" data (i.e., with replicas, with no standardized representation, etc.) goes far beyond technical questions such as the overall speed or performance of data management systems.

To better understand the impact of this problem [18], it is important to list and analyze the major consequences ofallowing the existence of "dirty" data in the repositories.These include, for example:

1) Performance degradation

DOI:03.LSCS.2013.6.68

2) Quality loss
3) Increasing operational costs
To avoid these problems, it is necessary to study thecauses of “dirty” data in repositories. A major cause is the presence of duplicates, quasi replicas, or near-duplicates in these repositories, mainly those constructed by the aggregation or integration of distinct data sources. The problem of detecting and removing duplicate entries in a repository is generally known as record deduplication [1] (but it is also referred to in the literature as data cleaning [2], record linkage [1][3][4], and record matching[5].
Often, while integrating data from different sources to implement a data warehouse, organizations become aware of potential systematic differences or conflicts. Such problems fall under the umbrella-term data heterogeneity [6][7] Data cleaning [8], or data scrubbing [9], refers to the process of resolving such identification problems in the data. We distinguish between two types of data heterogeneity: structural and lexical. Structural heterogeneity occurs when the fields of the tuples in the database are structured differently in different databases. For example, in one database, the customer address might be recorded in one field named, say, addr, while, in another database, the same information might be stored in multiple fields such as street, city, state, and zipcode.Lexical heterogeneity occurs when the tuples have identically structured fields across databases, but the data use different representations to refer to the same real-world object (e.g., StreetAddress = 44 W. 4th St. versus StreetAddress = 44 West Fourth Street).

II. LITERATURE SURVEY

Digital media has become a challenging problem for data administrators when volume of information is increased. Built on data gathered from different sources, data repositories such as those used by digital libraries and e-commerce brokers may present records with disparate structure.Data cleaning includes the process of parsing, data transformation, duplicate elimination and statistical methods. Record deduplication is the process of identifying and removing duplicate entries in a repository. It is also referred as data cleaning, record linkage and matching.

A. Supervised and Semisupervised Learning

The probabilistic model uses a Bayesian approach to classify record pairs into two classes, M and U. This model was widely used for duplicate detection tasks, usually as an application of the Fellegi-Sunter model. While the Fellegi-Sunter approach dominated the field for more than twodecades, the development of new classification techniques in the machine learning and statistics communities prompted the development of new deduplication techniques.

B. Active-Learning-Based Techniques

One of the problems with the supervised learning techniques is the requirement for a large number of training examples. While it is easy to create a large number of training pairs that are either clearly non duplicates or clearly duplicates, it is very difficult to generate ambiguous cases that would help create a highly accurate classifier. Based on this observation, some duplicate detection systems used active learning techniques [5] to automatically locate suchambiguous pairs. Unlike an “ordinary” learner that is trained using a static training set, an “active” learner actively picks subsets of instances from unlabeled data, which, when labelled, will provide the highest information gain to the learner. Sarawagi and Bhamidipaty [10] designed ALIAS, a learning-based duplicate detection system that uses the idea of a “reject region” to significantly reduce the size of the training set.

C. Distance-Based Techniques

Even active learning techniques require some training data or some human effort to create the matching models. In theabsence of such training data or the ability to get human input, supervised and active learning techniques are not appropriate. One way of avoiding the need for training data is to define a distance metric for records which does not need tuning through training data. Using the distance metric and an appropriate matching threshold, it is possible to match similar records without the need for training.

D.Rule-Based Approaches

A special case of distance-based approaches is the use of rules to define whether two records are the same or not.Rule-based approaches can be considered as distance-based techniques, where the distance of two records

is either 0 or 1. Wang and Madnick [11] proposed a rule-based approach for the duplicate detection problem. For cases in which there is no global key, Wang and Madnick suggest the use of rules developed by experts to derive a set of attributesthat collectively serve as a "key" for each record.

E.Unsupervised Learning

As we mentioned earlier, the comparison space consists of comparison vectors which contain information about the differences between fields in a pair of records. Unless some information exists about which comparison vectors correspond to which category (match, nonmatch, or possible match), the labelling of the comparison vectors in the training data set should be done manually. One way to avoid manual labelling of the comparison vectors is to use clustering algorithms and group together similar comparison vectors. The idea behind most unsupervised learning approaches for duplicate detection is that similar comparison vectors correspond to the same class.

The idea of unsupervised learning for duplicate detection has its roots in the probabilistic model proposed by Fellegi and Sunter . As we discussed, when there is no training data to compute the probability estimates, it ispossible to use variations of the Expectation Maximization algorithm to identify appropriate clusters in the data. Verykios et al. [12] propose the use of a bootstrapping technique based on clustering to learn matching models. The basic idea, also known as co training [13], is to use very few labelled data, and then use unsupervised learning techniques to appropriately label the data with unknown labels.

Several approaches to record deduplication have been proposed in recent years (Bilenko& Mooney, 2003; Dorneles et al.,2009; Carvalho et al., 2006; Carvalho, Laender, Gonçalves, & da Silva, 2008; Chaudhuri, Ganjam, Ganti, &Motwani, 2003;Cohen & Richman, 2002; Tejada, Knoblock, & Minton, 2001). Most of these works focus on the deduplication task in the contextof integration of relational databases. Few automatic approaches, if any, have been specifically developed for the realmof digital libraries or in a more general sense, for bibliographic metadata records.

A general framework of fuzzylogical-based similarity measures [14] based on equalities that arederived from residual implication functions is proposed. Then, amodel that allows us to learn the parametric similarity measuresis introduced. This is achieved by an online learning algorithmwith an efficient implication-based loss function.

Supervised record linkage methods often require a clerical review to gain informative training data. Active learning [15][16] means to actively prompt the user to label data with special characteristics in order to minimise the review costs. We conducted an empirical evaluation to investigate whether a simple active learning strategy using binary comparison patterns is sufficient or if string metrics together with a more sophisticated algorithm are necessary to achieve high accuracies with a small training set.

F. Concluding Remarks

There are multiple techniques for duplicate record detection. We can divide the techniques into two broad categories: ad hoc techniques that work quickly on existing relational databases and more "principled" techniques that are based on probabilistic inference models. While probabilistic methods outperform ad hoc techniques in terms of accuracy, the ad hoc techniques work much faster and can scale to databases with hundreds of thousands of records. Probabilistic inference techniques are practical today only for data sets that are one or two orders of magnitude smaller than the data sets handled by ad hoc techniques. A promising direction for future research is to devise techniques that can substantially improve the efficiency of approaches that rely on machine learning and probabilistic inference.

III. MODULE DESCRIPTION

Figure 1.gives an overview of the different componentsused within the application to conduct experiments. Theframework is implemented in .NET framework, which makes it easy toextend. Following is the detail description of each component:

A. Experimental Datasets

In our experiments, we used two real data sets commonlyemployed for evaluating record deduplication approaches [13][14], which are based on real data gathered from theweb.

The first real data set, the Cora Bibliographic data set, is acollection of 1,295 distinct citations to 122 computer sciencepapers taken from the Cora research paper search engine.These citations were divided into

multiple attributes (authornames, year, title, venue, and pages and other info) by aninformation extraction system.
The second real data set, hereafter named the Restaurantsdata set, contains 864 entries of restaurant names andadditional information, including 112 duplicates that wereobtained by integrating records from Fodor and Zagat'sguidebooks. We used the following attributes from this dataset: (restaurant) name, address, city, and specialty.

B. Cleaning and Standardisation

Data standardization [17] refers to the process of standardizing the information represented in certain fields to a specific content format. This is used for information that can be stored in many different ways in various data sources and must be converted to a uniform representation before the duplicate detection process starts. Without standardization, many duplicate entries could be erroneously designated as non duplicates based on the fact that common identifying information cannot be compared.
Most real-world data are dirty and contain noisy, incomplete and incorrectly formatted information, a crucial first step in any record linkage or deduplication project is data cleaning and standardization. It has been recognized that a lack of good quality data can be one of the biggest obstacles to successful record linkage. The main task of data cleaning and standardization is the conversion of the raw input data into well defined, consistent forms, as well as the resolution of inconsistencies in the way information is represented and encoded.

Figure 1. System Block Diagram

C. Indexing

The aim of the indexing step is to reduce thislarge number of potential comparisons by removing asmany record pairs as possible that correspond to nonmatches.The traditional record linkage approach [3] has employed an indexing technique commonlycalled *blocking*, which splits the databases into nonoverlappingblocks, such that only records within eachblock are compared with each other. A blocking criterion,commonly called a *blocking key* (the term used in thispaper), is either based on a single record field (attribute),or the concatenation of values from several fields.

D. Record Pair Comparison

The indexing step generates pairs of candidate records that are compared in detail in the comparison step using a variety of comparison functions appropriate to the content of the record fields (attributes). Approximate string comparisons, which take (typographical) variations into account, are commonly used on fields that for example contain name and address details, while comparison functions specific for date, age, and numerical values are used for fields that contain such data. Several fields are normally compared for each record pair, resulting in a vector that contains the numerical similarity values calculated for that pair.
The similarity function is used to compare the field (attribute) values of record pairs. These similarity functions re-turn a numerical value between 0 (total dissimilarity) and 1(exact match). It is possible to adjust these values by setting agreement and disagreement weights, as well as a specialvalue that will be returned if one or both of the comparedvalues is/are empty. The similarity weights calculated foreach compared record pair will be stored in a weight vector,to be used for classifying record pairs in the next step.

E. Feature Vector

It contains several record pair classifiers, both supervised and unsupervised techniques. The traditional Fel-legiSunter' classifier requires manual setting of two thresholds [13], while with the supervised Optimal Threshold' classifier it is assumed that the true match status for all compared record pairs is known, and thus an optimal threshold can be calculated based on the corresponding summedweight vectors.

F. Genetic Programming

GP [18] evolves a population of length-free data structures, also called records, each one representing a single solution to a given problem. During the generating process, the records are handled and modified by genetic operations such as reproduction, crossover, and mutation, in an iterative way that is expected to spawn better records (solutions to the proposed problem) in the subsequent generations.

In this work, the GP generation wise process is guided by a generational evolutionary algorithm. This means that there are well defined and distinct generation cycles. It can adopt this approach since it captures the basic idea behind several generation wise algorithms. The algorithm steps are the following:

1. Initialize the population (with random or user provided records).
2. Evaluate all records in the present population, assigning a numeric rating or fitness function to each record.
3. If the termination criterion is satisfied, then execute the last step. Otherwise continue.
4. Select the best n individuals into the next generation population.
5. Select m individuals that will compose the next generation with the best parents.

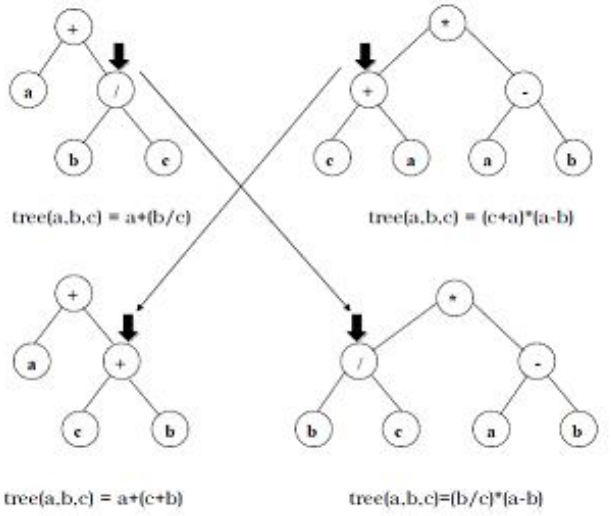

Figure 2. Random Subtree Crossover

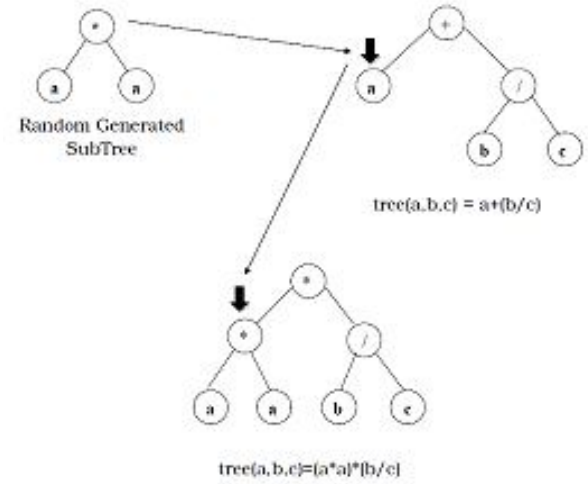

Figure 3. Random Subtree Mutation

6. Apply the genetic operations to all records selected. Their children will compose the next population. Replace the existing generation by the generated population and go back to Step 2.
7. Present the best record(s) in the population as the output of the evolutionary process.

G. Basic Genetic operations

In GA [18], a candidate solution for a specific problem is called an individual or a chromosome and consists of a linear list of genes. Each individual represents a point in the search space, and hence a possible solution to the problem. A population consists of a finite number of individuals. Each individual is decided by an evaluating mechanism to obtain its fitness value. Based on his fitness value and undergoing genetic operators, a new population is generated iteratively with each successive population referred to as a generation. The GAs uses three basic operators (reproduction, crossover, and mutation) to manipulate the genetic composition of a population.

Reproduction is a process by which the most highly rated individuals in the current generation are reproduced in the new generation.

The crossover operator produces two offspring's (new candidate solutions) by recombiningthe information from two parents. There are two processing steps in this operation. In the first step, a given number ofCrossing sites are selected uniformly, along with the parentindividual at random. In the second step, two new individuals are formed by exchanging alternatepairs of selection between the selected sites.In a GP evolutionaryprocess, two parent trees are selected according to amatching (or pairing) policy and, then, a random subtreeis selected in each parent. Child trees are the result from theswap of the selected subtrees between the parents, asillustrated in Figure 2.

Mutation is a random alteration of some gene values in an individual. The allele of each gene is a candidate for mutation, and its function is determined by the mutation probability. Every solutiontree resulting from the crossover operation has an equalchance of suffering a mutation process. In a GP treerepresentation, a random node is selected and thecorresponding subtree is replaced by a new randomlycreated subtree, as illustrated in Figure 3.

IV. MODELING THE RECORD DEDUPLICATION PROBLEM WITH GP

When using GP (or even some other evolutionary technique)[18] to solve a problem, there are some basic requirements that must be fulfilled, which are based on the data structureused to represent the solution [8]. In our case, we havechosen a tree-based GP representation for the deduplicationfunction, since it is a natural representation for this type offunction. These requirements are the following:

1. All possible solutions to the problem must berepresented by a tree, no matter its size.
2. The evolutionary operations applied over each individualtree must, at the end, result into a valid tree.
3. Each individual tree must be automatically evaluated.

For Requirement 1, it is necessary to take into considerationthe kind of solution we intend to find. In the recorddeduplication problem, we look for a function thatcombines pieces of evidence. In our approach, each piece of evidence (or simply"evidence") E is a pair <attribute; similarity function>that represents the use of a specific similarity function overthe values of a specific attribute found in the data beinganalyzed. For example, if we want to deduplicate adatabase table with four attributes (e.g., forename, surname,address, and postalcode) using a specific similarity function (e.g., the Jaro function), we would have the followinglist of evidence: E1<name; Jaro>, E2<surname; Jaro>,E3<address; Jaro>, and E4<postalcode; Jaro>.

The tree input is a set of evidence instances, extractedfrom the data being handled, and its output is a realnumber value. This value is compared against a replicaidentification boundary value as follows: if it is above theboundary, the records are considered replicas; otherwise,the records are considered distinct entries. It is important tonotice that this classification enables further analysis,especially regarding the transitive properties of the replicas.This can improve the efficiency of clustering algorithms,since it provides not only an estimation of thesimilarity between the records being processed, but also ajudgment of whether they are replicas or not. After doing these comparisons for all candidate recordpairs, the total number of correct and incorrect identifiedreplicas is computed. This information is then used by themost important configuration component in our approach: the fitness function.The fitness function is the GP component that isresponsible for evaluating the generated individuals alongthe evolutionary process. If the fitness function is badlychosen or designed, it will surely fail in finding a goodindividual. In the experiments presented in this paper, wehave

used the F1 metric as our fitness function. The F1metric harmonically combines the traditional precision (P)and recall (R) metrics commonly used for evaluatinginformation retrieval systems , as defined below:

$$P = \frac{NumberOfCorrectlyIdentifiedDuplicatedPairs}{NumberOfIdentifiedDuplicatedPairs}$$

$$R = \frac{NumberOfCorrectlyIdentifiedDuplicatedPairs}{NumberOfTrueDuplicatedPairs}$$

$$F1 = \frac{2 \times P \times R}{P + R}.$$

The best tree (the best individual extracted from the last generation) for the experiment with the Cora data set, when we used the string distance similarity function, was

GPCoraStrDist( )=((e+d)*d)+(c+(a*2))+d

where a, c, d, and e correspond to evidence defined on the attributes author names, title, venue, and pages and other info, respectively.

In this function, the string distance similarity function was applied to four out of five attributes. Only the attribute year was not used. This happened because the cosine similarity function, when applied to dates, is not able to properly measure the "distance" between them. By analyzing the function, we can say that the attributes venue (used in evidence d) and author names (used in evidence a) are those that play the most important role for identifying replicas in the Cora data set, since they have higher weights, i.e., evidence a is multiplied by 2 and evidence d appears three times.

The best tree for the experiment with the Restaurants data set, when we used the cosine similarity function, was

GPRestaurantCosine( )=((b+(b+d)*a)+2

where a, b, and d correspond to evidence defined on the attributes name, address, and specialty, respectively.

In this function, we notice that only the attribute city was not used as evidence. It is also important to notice that address (used in evidence b) was the attribute which received more weight in the function. This may be explained because the cosine similarity function is more discriminative when applied to long multiple string attributes, like addresses, than when applied to short strings (e.g., dates or short names).

V. Conclusions and Future Work

As database systems are becoming more and morecommonplace, data cleaning is going to be thecornerstoneforcorrecting errors in systems which are accumulatingvast amounts of errors on a daily basis. Despite the breadthand depth of the presented techniques, we believe that thereis still room for substantial improvement in the currentstate-of-the-art.

In this paper, we presented a GP-based approach torecord deduplication. Our approach is able to automaticallysuggest deduplication functions based on evidence presentin the data repositories. The suggested functions properlycombine the best evidence available in order to identifywhether two or more distinct record entries are replicas (i.e.represent the same real-world entity) or not.GP with indexing increases the performance tremendously to record deduplication.

As future work, we intend to conduct additional research in order to extend the range of use of our GPbased approach to record deduplication. For accomplishing this, we plan experiments with data sets from other domains. Other future work includesthe implementation of further recently developed newindexing techniques, new similarity functions.

References

[1] N. Koudas, S. Sarawagi, and D. Srivastava, "Record Linkage: Similarity Measures and Algorithms," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 802-803, 2006.

[2] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and Efficient Fuzzy Match for Online Data Cleaning," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 313-324, 2003.

[3] I. Bhattacharya and L. Getoor, "Iterative Record Linkage for Cleaning and Integration," Proc. Ninth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery, pp. 11-18, 2004. DE CARVALHO ET AL.: A GENETIC PROGRAMMING APPROACH TO RECORD DEDUPLICATION 411.

[4] I.P. Fellegi and A.B. Sunter, "A Theory for Record Linkage," J. Am.Statistical Assoc., vol. 66, no. 1, pp. 1183-1210, 1969.

[5] V.S. Verykios, G.V. Moustakides, and M.G. Elfeky, "A Bayesian Decision Model for Cost Optimal Record Matching," The Very Large Databases J., vol. 12, no. 1, pp. 28-40, 2003.
[6] A. Chatterjee and A. Segev, "Data Manipulation in Heterogeneous Databases," ACM SIGMOD Record, vol. no. 4, pp. 64-68, Dec. 1991.
[7] IEEE Data Eng. Bull., S. Sarawagi, ed., special issue on data cleaning, vol. 23, no. 4, Dec. 2000.
[8] J. Widom, "Research Problems in Data Warehousing," Proc. 1995 ACM Conf. Information and Knowledge Management (CIKM '95), pp. 25-30, 1995.
[9] S. Sarawagi and A. Bhamidipaty, "Interactive Deduplication Using Active Learning," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '02), pp. 269-278, 2002.
[10] Y.R. Wang and S.E. Madnick, "The Inter-Database Instance Identification Problem in Integrating Autonomous Systems," Proc. Fifth IEEE Int'l Conf. Data Eng. (ICDE '89), pp. 46-55, 1989.
[11] V.S. Verykios, A.K. Elmagarmid, and E.N. Houstis, "Automating the Approximate Record Matching Process," Information Sciences, vol. 126, nos. 1-4, pp. 83-98, July 2000.
[12] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," COLT '98: Proc. 11th Ann. Conf. Computational Learning Theory, pp. 92-100, 1998.
[13] M. Bilenko and R.J. Mooney, "Adaptive Duplicate Detection Using Learnable String Similarity Measures," Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 39- 48, 2003.
[14] S. Tejada, C.A. Knoblock, and S. Minton, "Learning Domain- Independent String Transformation Weights for High Accuracy Object Identification," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 350-359, 2002.
[15] Hoel Le Capitaine ,"A Relevance-Based Learning Model of Fuzzy Similarity Measures " VOL. 20, NO. 1, FEBRUARY 2012.
[16] M. Sariyarî, A. Borg, K. Pommerening Active learning strategies for the deduplication of electronic patient data using classification trees Journal of Biomedical Informatics 45 (2012) 893–900.
[17] A K Elmagarmid, P G Ipeirotis, V S Verykios ,Duplicate Record Detection: A Survey,IEEE Transactions on Knowledge and Data Engineering (2007) Volume: 19, Issue: 1, Publisher: IEEE Computer Society.
[18] Moise´s G. de Carvalho, Alberto H.F. Laender, Marcos Andre´ Gonc¸alves, and Altigran S. da Silva, A Genetic Programming Approach to Record Deduplication, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012